Failure and communication in a synchronized multi-drone system

S. Bereg[1], J.M. Díaz-Báñez[2], Paul Horn[3], Mario A. Lopez[4], and Jorge Urrutia[5]

[1] Department of Computer Science, University of Texas at Dallas, USA. Partially supported by NSF award CCF-1718994.
[2] Department of Applied Mathematics II, University of Seville, SPAIN. Partially supported by Spanish Ministry of Economy and Competitiveness project MTM2016-76272-R AEI/FEDER,UE and European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant #734922.
[3] Department of Mathematics, University of Denver, USA. Partially supported by Simons Collaboration Grant #525039.
[4] Department of Computer Science, University of Denver, USA. Partially supported by a University of Denver John Evans Award.
[5] Instituto de Matemáticas, Universidad Nacional Autónoma de México, México. Partially supported by PAPIIT Grant IN 102117, UNAM.

Abstract. A set of n drones with limited communication capacity is deployed to monitor a terrain partitioned into n pairwise disjoint closed trajectories, one per drone. In our setting, there is a communication link between two trajectories if they are close enough, and drones can communicate provided they visit the link at the same time. Over time, one or more drones may fail and the ability to communicate and stay connected decreases. In this paper we study two properties related to communication: isolation and connectivity. First, we provide efficient algorithms, both centralized and decentralized, for determining the connected components induced by the set of surviving drones. Second, we study isolation and connectivity under a probabilistic failure model and show that, in the case of grids, the system is quite robust in the sense that it can tolerate a large probability of failure before drones become isolated and the system loses full connectivity.

Keywords: Unmanned Aerial Vehicles · Synchronized Communication System · Communication Graph · Connectivity · Probalilistic model.

1 Introduction

Teams of *Unmanned Aerial Vehicles* (UAVs), colloquially known as drones, are becoming a trend in the last few years for their use in a wide variety of applications such as area monitoring, precision agriculture, search and rescue, exploration and mapping, and delivery of products, to name a few; see [10, 12, 11] and references therein for a comprehensive survey on the topic. The coordination of a team of autonomous vehicles enables the execution of tasks that no individual autonomous vehicle can accomplish on its own, and thus there has been an increasing interest in studying teams of drones that cooperate with each other. In such multi-drone systems a desired collective outcome arises from the interaction of the drones with each other and with their environment, via a set of installed sensors and communication devices.

The problems raised in this paper assume the framework recently proposed in [7]. A partition of a terrain to be covered is given and every drone is assigned a different section of the partition. Each drone travels on a fixed closed trajectory while performing a prescribed task, such as monitoring its assigned area. In order to allow cooperation, each drone needs to communicate periodically with other drones. Since the UAVs have a limited communication range, two of them need to be in close proximity of each other in order to communicate. In [7] the authors presented a framework to survey a terrain in the scenario described above. As an abstraction, they considered a model in which each drone is modeled by a single point that flies on a unit circle at constant speed, and this speed is the same for all the drones. They assume, w.l.o.g., that one time unit is the time required by a robot to complete a tour of a circle. These circles may intersect at a single point but do not cross. The communication between two robots can take place if their corresponding circles touch, and it is carried out at the point of intersection. They also showed how to generalize the results to a more realistic model. In [7] it is assumed that the unit disk graph defined by the given set of circles (trajectories) is connected, they call it the *communication graph*.

The main problem addressed in [7] is to obtain a synchronization schedule, that is, to assign a starting position and travel direction to each trajectory so that if n drones follow this schedule, every pair of them traveling in two adjacent circles pass through the intersection point of their trajectories at the same time. A set of trajectories with a synchronization schedule conform a *synchronized communication system* (SCS) [7]. In the same paper, the authors also discuss necessary and sufficient conditions for the existence of a synchronization schedule. For an illustration see Figure 1 and related video[6]. Note that although not every pair of robots can communicate directly, a robot may relay a message to another robot through a sequence of intermediate message exchanges.

If the system is synchronized, as described above, a robot can easily detect the failure of a neighboring robot. If a robot d_i in trajectory C_i arrives at the communication point between C_i and another trajectory C_j, and it fails to meet another robot, it will assume that the robot in C_j is no longer functional. Under such circumstances, a reasonable strategy is for d_i to switch to C_j at this point and take over the task of the missing robot. In [7], this strategy is called the *shifting strategy*. Under the shifting strategy, an undesirable phenomenon, known as *isolation*, may occur. A drone is *isolated* if it fails permanently to meet other drones. The three black drones in Figure 1(b) never meet, and thus they are isolated. A *ring* is the closed path followed by an isolated drone. Each ring is composed of sections of various trajectories and has a direction of travel determined by the direction of movement in the participating trajectories. Each

[6] `https://www.youtube.com/watch?v=T0V6tO8OHOI`

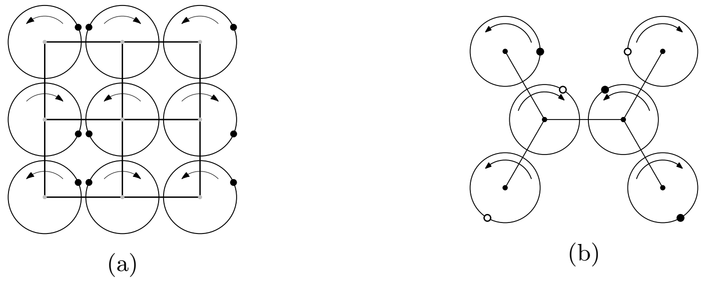

Fig. 1: Examples of synchronized communication systems. The robots in the SCS are represented by solid black points. (a) The communication graph is a grid. (b) The communication graph is a tree. If the white drones leave the system, the black drones become isolated.

section of a trajectory between two consecutive link positions participates in exactly one ring, thus the rings in an SCS are pairwise disjoint. The number of rings and its length depends on the communication graph. Figure 2 illustrates some examples. See [2] for a study on rings and the isolation phenomenon.

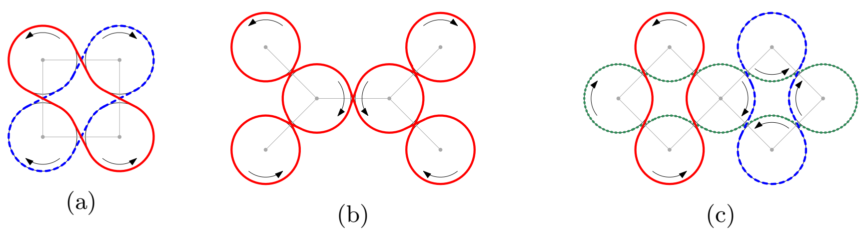

Fig. 2: SCSs with two rings (a); one ring (b) and three rings (c).

In [7], neighboring circles are assigned opposite travel directions (clockwise and counterclockwise) so as to enable the shifting strategy. From now on we work with SCSs where every pair of neighboring circles have opposite travel directions and, consequently, the communication graph is bipartite. Under this model, our main contributions are the following:

1. *Connected components*. Consider a system in which some drones may have failed. Two drones belong to the same connected component if they can exchange messages, possibly through a sequence of intermediaries. We provide efficient algorithms for computing the connected components of the system, both centralized (where a central server is privy of a snapshot of the system, Theorem 2) and decentralized (using only the information that drones can gather while flying and meeting other drones, Theorem 4). For the case of grids, the required flying time can be proportional to the number of trajectories, and this bound is tight (Theorem 3).

2. *Probabilistic failure model.* We address the robustness of a system in which drones survive with probability p, and study two properties: full connectivity and drone isolation. For $t \times t$ grids, we establish sharp thresholds for the existence of isolated drones (Theorem 5) and connectivity (Theorem 6). These results show that the system is extremely robust to random failure as these thresholds are $o(1)$ as $t \to \infty$. For general grids, we provide less sharp results (Theorems 7 and 8).

2 Related work

There is a vast literature related to communication strategies for a team of robots monitoring a given area. Our scenario shares similarities with work on patrolling agents [9] where the drones patrol along predefined paths making observations and synchronize with their teammates during a very limited time to share data. Typically, research has focused mostly on construction and validation of working systems, rather than more general and formal analysis of problems. In this paper, we study some algorithmic and probabilistic problems related to communication in the particular framework of a synchronized communication system (SCS) proposed in [7].

Recently, the study of stochastic UAV systems has attracted considerable attention in the field of mobile robots. This approach has several advantages such as shorter times to complete tasks, cost reduction, higher scalability, and more reliability, among others [13]. In a pure random mobility model, each node randomly selects its direction, speed, and time independently of other nodes. Some models include random walks, random waypoints or random directions. See [3] for a comprehensive survey. The framework assumed in this paper is not a pure random walk model but the use of stochastic strategies in the shifting protocol generates random walks [4]. In the same work, the authors evaluate both the coverage and communication performance of a SCS and show the validity of two random strategies compared with the deterministic one.

On the other hand, several algorithmic and combinatorial problems have been studied within a SCS using the deterministic shifting strategy. In [2, 1] the authors propose various quality measures for a synchronized system regarding the resilience of a network in the presence of failures. Computing these measures leads to interesting combinatorial and algorithmic problems.

3 Computing the connected components

Assume we have a SCS where a subset of drones left the system and the surviving drones apply the shifting protocol. We call the resulting system a *partial* SCS. While some pairs of drones may communicate directly, communication between other pairs may rely on passing information through other drones; in some cases communication between drones may be impossible. We define the *drone communication graph* G_D as the graph whose vertices are the drones, two of which are adjacent if the corresponding drones communicate directly at some point in time. The connected components of this graph identify which sets of drones can, directly or indirectly, communicate with each other. It is easy to see that communication through other robots can sometimes be faster than direct communication, e.g. it may take a long time for two drones to communicate directly. In this section we show how to compute the connected components in the drone communication graph under two models of computation:

1. **Centralized:** Suppose a central server contains the full information of a SCS, including the set of drone trajectories and the initial locations of the drones. How can the connected components of the SCS be found efficiently?
2. **Decentralized:** Suppose the drones themselves can pass messages when they pass by each other. How can they determine the other drones in their connected component, and how quickly can this be accomplished?

Note that in the second case, the drones do not know how many other drones are active or where they are; they merely learn what drones are active as they meet other drones and exchange information. For that reason, the complexity of both problems is different. The complexity of our algorithm in the first case, is the number of steps the central server needs to be computed, while in the second case it is the *flying time* of the drones before each drone knows its connected component.

Nonetheless, we show that for the $s \times t$ grid both problems can be solved with highly efficient algorithms. The key notion for our results is the use of the *token graph* introduced in [1]. We assume that at time 0 each drone d_i holds a token t_i. This establishes a bijection between the drones and the tokens. When two drones meet, they exchange their tokens. The token graph G_T of a drone system is the graph whose vertices are the tokens, two of which are adjacent if at some time the corresponding tokens are exchanged. Note that each token t_i stays in the same ring of drone d_i. Thus, the token graph can also be defined using drones as vertices where two drones are adjacent if they encounter each other in a system where only two drones exist. We have the following result.

Theorem 1. *Two drones of G_D are in the same component if and only if the corresponding tokens are in the same component in G_T.*

In the case of an $s \times t$ grid with only two drones, the drones encounter each other if and only if they are in the the same row or the same column [1]. In this case the token graph can be viewed as the graph where the vertices are the drones and two drones are adjacent if at any point in time they are in the same row or column. If this happens, except when transitioning from one trajectory to another, the said drones will always be in the same row (both moving down or up) or column (both moving left or right) [1]. We call it an *RC-graph.*

Theorem 2. *The connected components in G_D can be computed in polynomial time in the centralized model. Furthermore, they can be computed in linear time in the $s \times t$ grid.*

Proof. The token graph can be computed in polynomial time and, for the $s \times t$ grid it can be computed in linear time [1]. The connected components in the token graph can be computed in linear time using breath-first search. By Theorem 1, this yields the connected components in G_D in the centralized model. For a drone system on a grid, the token graph is the RC-graph. The RC-graph and its connected components can be computed in linear time.

3.1 Decentralized Computation

The goal in the decentralized model is for each drone d_i to compute $C(d_i)$, its connected component in G_D. We use the following algorithm. Each drone d_i maintains a list $L(d_i)$ of some drones from $C(d_i)$. Initially, we set $L(d_i) := \{d_i\}$. When drones d_i and d_j meet at some time, they replace both $L(d_i)$ and $L(d_j)$ with the union $L(d_i) \cup L(d_j)$. It is clear that if we follow this protocol long enough, all drones will know their connected components (that is $L(d_i) = C(d_i)$ for all i.) Our goal is to give bounds for the running time of this approach.

We emphasize, again, that the time measured here is actually flying time of the drones, i.e. how long do the drones have to fly until they each, individually, know the other drones own components. We ignore, then, the computation of the set unions involved as this is negligible compared with the actual flying time from one communication point to another. We further assume that a unit time is needed to navigate a trajectory. We begin with sharp bounds for the problem on a $t \times t$ grid.

Theorem 3. *On the $t \times t$ grid, at time $t \cdot (t-1)$ we have that $L(d) = C(d)$ for all drones d. Furthermore, there are drone configurations that require $\Omega(t^2)$ time until $L(d) = C(d)$ for all drones d.*

Proof. We use the idea of tokens. At the beginning (time 0), each drone d_i holds token t_i; recall that when two drones encounter each other they exchange tokens (along with taking the union of their respective lists). Let $d(i, m)$ denote the drone holding token t_i at time m. Thus, $d(i, 0) = d_i$. Note that $d(i, m)$ is always in the same component as drone d_i as it holds t_i due to a sequence of interactions with other drones, each passing t_i to the next drone of the sequence. Moreover, $L(d(i, m)) \subseteq L(d(i, m'))$ if $m \leq m'$ as whenever tokens are exchanged, the lists are passed along.

Fix an (arbitrary) drone d_0 and consider any drone d_k in $C(d_0)$. Let $d_0, d_1, \ldots,$ d_k be the shortest path between d_0 and d_k in the token graph. By the construction of the token graph as an RC-graph, it is easy to see that the diameter of the token graph is at most $t-1$, and in particular $k \leq t-1$. Note that t_i and t_{i+1} are in the same row or column, and hence drones holding them meet within time t. This implies that, for instance, $d_1 \in L(d(0, t))$ at time t as when the drones with token t_0 and t_1 meet, the label d_1 is passed to the drone holding token t_0. Inductively, it follows $d_i \in L(d(0, i \cdot t))$ at time $i \cdot t$: the label d_i is given to the drone carrying token t_{i-1} in the first time t, then to the drone carrying token t_{i-2} in the next time t, until it at last is passed to the drone hoping token t_0. This shows that $L(d(0, t(t-1))$ is complete at time $k \cdot t \leq (t-1)t$ and as d_0 is arbitrary, this completes the proof.

We now provide a set of drones $\{d_1, \ldots, d_k\}$ which show this time can be quadratic. For this set of drones, $d_1 \notin L(d(k,m))$ until time $m = \Omega(t^2)$. The construction involves a set of drones $\{d_1, \ldots, d_k\}$ on the $t \times t$ grid satisfying the following conditions:

1. d_i and d_{i+1}, for $i = 1, 2, \ldots, k$ share the same row or column, and there are no rows or columns with more than two drones.
2. The distances $d_{i,i+1}$ between d_i and d_{i+1} are decreasing, for $i = 1, 2, \ldots, k$.
3. The polygonal chain formed by the union of the segments connecting d_i to d_{i+1} is a *spiral* polygonal chain; see Figure 3.
4. Drones on the same column move in opposite directions (clockwise and counterclockwise) along their ring, while drones in the same row move in the same direction.

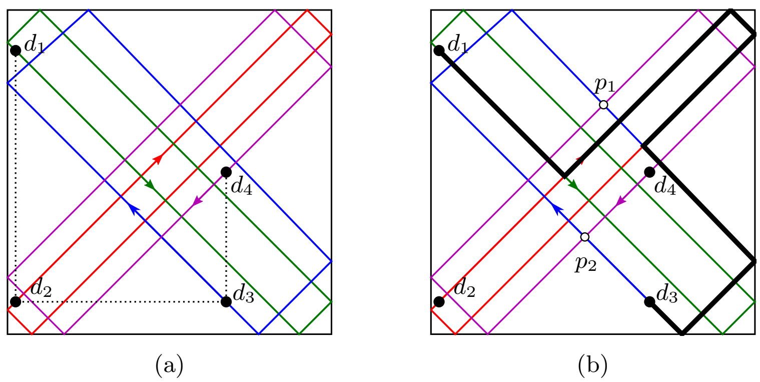

Fig. 3: (a) Drones are arranged in a spiral polygonal chain. (b) The bold line represents the propagation of the label d_1 through the system for times $\leq t$. Drones holding tokens t_3 and t_4 meet at p_1 and p_2.

The key observation is that the drone holding token t_i will only meet the drones holding token t_{i+1} and t_{i-1} and by placing the drones carefully, the intersections will be set up so that the label of d_1 will only propagate through a small number of consecutive drones in time t.

Consider four consecutive drones in $\{d_1, \ldots, d_k\}$; without loss of generality assume these are d_1, d_2, d_3 and d_4. We claim that at time t, the only elements in $\{d_1, \ldots, d_k\}$ with $d_1 \in L(d(i,t))$ are $i = 1, 2, 3$. To see this, observe that since $d_{1,2} > d_{2,3}$, t_2 meets t_3 the first time before it meets t_1. The label d_1 is thus added to the list of the drone holding token t_3 during the second time the drones holding tokens of t_2 and t_3 meet. At this point t_3 has already swapped with t_4 twice. Hence $d_1 \notin L(d(4,t))$ at time t.

Figures 3a and 3b illustrate first this setup and then the process itself. Figure 3 illustrates how the threshold of knowledge of drone d_1 moves forward through the process Note it never moves from the drone holding token t_1 directly to, say, that holding token t_3 as even though the rings of these drones intersect, drones holding these tokens never directly communicate due to the timing. It follows that to reach the drone holding token t_{2i} label d_1 will take $t(i+1)$ time. □

For a general system, a similar argument can be used to prove the following.

Theorem 4. *Consider a general system of N drones on n trajectories and ring lengths $r_1, r_2, \ldots, r_k$. Then at time $N \cdot \max\{lcm(r_l, r_m) : l \neq m\}$, $L(d_i) = C(d_i)$ for all drones $d_i, i = 1, \cdots, N$.*

4 Communication within a probabilistic failure model

In this section we study the connectivity of G_D under random failure. We prove sharp thresholds for the properties of containing an isolated vertex and for connectivity. We remark that our results are quite similar to those for the well known Erdős-Rényi random graph [8], but our setting differs in two crucial ways. First the 'host graph' (which can be thought of as the RC-graph for the full system) is not complete; nor is the resulting random G_D a subgraph of the full G_D as which drones directly communicate within a subsystem differs from that of the full. Second, while most work generalizing results of the Erdős-Rényi graph to more general host graphs (see, eg. [5, 6]) takes a random set of edges, we actually take a random set of vertices. A side effect is that the properties we study are not monotone; additional drones surviving may break these properties.

Theorem 5. *Consider a full drone system in the $t \times t$ grid, where drones survive with probability p. Let $\mathcal{I}$ denote the event that some drone is isolated. Fix an arbitrary $\varepsilon > 0$.*

(a) If $p = (1+\varepsilon)\frac{\ln t}{2t}$ then as $t \to \infty$, then $\mathbb{P}(\mathcal{I}) \to 0$.
(b) If $p = (1-\varepsilon)\frac{\ln t}{2t}$ then as $t \to \infty$, then $\mathbb{P}(\mathcal{I}) \to 1$.

Proof. For (a), note that there are t^2 drone locations and in order for a drone to be isolated it must survive and all others in its row and column must fail. Hence the expected number of isolated drones is

$$t^2 p(1-p)^{2t-1} \leq t^2 p e^{-p(2t-1)} = (1+\varepsilon)\frac{t \ln t}{2} \exp\left(-(1+\varepsilon)\frac{(2t-1)}{2t}\ln(t)\right) \to 0,$$

where we note that for t sufficiently large $(1+\varepsilon)\frac{(2t-1)}{2t} > 1$, so that the exponential term is $O(t^{-(1+\varepsilon')})$. (a) then follows by Markov's inequality.

For (b), note that the expected number of isolated drones in this situation is

$$t^2 p(1-p)^{2t-1} \leq t^2 p e^{-p(2t-1)} = (1-\varepsilon)\frac{t \ln t}{2} \exp\left(-(1-\varepsilon)\frac{(2t-1)}{2t}\ln(t)\right) \geq t^{\varepsilon/2},$$

assuming that t is sufficiently large.

For (b), then it suffices, by Chebyschev's inequality, to show that if X is the number of isolated drones in the system, to show that $\mathrm{Var}(X) = o(\mathbb{E}[X]^2)$. Note that X can be written as $\sum_{(i,j)\in[t]^2} X_{i,j}$, where $X_{i,j}$ is the event that the drone in the $(i,j)th$ position is isolated. Then

$$\mathrm{Var}(X) \le \mathbb{E}[X] + \sum_{(i,j)\neq(k,l)\in[t]^2} \Big(\mathbb{E}[X_{i,j}X_{k,l}] - \mathbb{E}[X_{i,j}]\mathbb{E}[X_{k,l}] \Big).$$

We bound the sum. If $i = k$ or $j = l$, then $\mathbb{E}[X_{i,j}X_{k,l}] = 0$, and $\mathbb{E}[X_{i,j}] = \mathbb{E}[X_{k,l}] = p(1-p)^{2t-1}$. As the covariance terms being sum are negative these terms can be discarded for upper bounding the variance. For the other terms, where (i,j) and (k,l) are different in both coordinates, $\mathbb{E}[X_{i,j}X_{k,l}] = p^2(1-p)^{4t-4}$ and there are at most t^4 terms of this type and these summands contribute at most

$$t^4 \left(p^2(1-p)^{4t-4} - p^2(1-p)^{4t-2}\right) = \mathbb{E}[X]^2((1-p)^{-2} - 1) = o(\mathbb{E}[X]^2),$$

where the last equality follows from the form of p. Hence, by Chebyschev's inequality $X \sim \mathbb{E}[X]$ with probability tending to one, and thus are isolated drones. □

Remark 1. Note that for (a), if suffices that $p \ge (1+\varepsilon)\frac{\log t}{t}$ – this follows as the expected number is decreasing in as p increases (assuming that $p \ge \frac{1}{2t-1}$.) Extending the lower bound works so long as the expected number of isolated drones tends to infinity.

Remark 2. Theorem 5(a) implies that, even for fairly small p, the number of isolated drones is 0 with high probability. At the threshold, the number of surviving drones is only $O(t \log t)$, while $t^2 - O(t \log t)$ drones fail in this case. This should be compared with the 1-isolation resilience of the grid, the miniminum number of drones whose failure can result in an isolated drone, which is $O(t)$ [1].

Theorem 6. *Consider a full drone system in the $t \times t$ grid, where drones survive with probability p. Let $\mathcal{C}$ denote the event that the system of drones is connected (that is, all drones can communicate with one another). Fix an arbitrary $\varepsilon > 0$.*

(a) If $p = (1+\varepsilon)\frac{\ln t}{2t}$ then as $t \to \infty$, then $\mathbb{P}(\mathcal{C}) \to 1$.
(b) If $p = (1-\varepsilon)\frac{\ln t}{2t}$ then as $t \to \infty$, then $\mathbb{P}(\mathcal{C}) \to 0$.

Proof. Note that (b) follows directly from Theorem 5, as if there is an isolated drone (and more than one drone, as there is at such a p with high probability) then the system is not connected.

We proceed to prove (a). We have already shown that when $p = (1+\varepsilon)\frac{\ln t}{t}$ that there are no components of size 1. We still need to show there is a unique component. To do this, we study a modified breadth first search in the RC-graph, introduced in the previous section. Recall, that performing a breadth-first search in the RC-graph (where vertices are drones and they are joined if they in the same row or column) reveals the connected component of a vertex.

To show that there is precisely one component in this setting, we study a slightly modified tree finding algorithm. An *exposing tree* inside of a component is a rooted tree generated as follows: Choose an initial root vertex (drone) to explore. Add all vertices in its row and column to a queue. Now, each vertex in the queue is iteratively explored. When a vertex is explored, vertices in their row or column are added to the queue *if either their row or column is different from those already added to the queue.* Since every vertex being explored was added to the queue it shares either a row or column with one of the other vertices previously explored, and each vertex is responsible for 'exposing' a new row or new column (with the initial vertex responsible for exposing both.) The set of explored vertices forms the exposing tree.

Generating an exposure tree ends with a subset of a connected component which is both non-empty and possibly proper – but vertices in the component and not in the tree share both a row and a column with vertices in the tree. It also ends with a drone from each of some j columns and k rows (where j and k are determined by the process) and $j+k-1$ vertices. Furthermore, the process ending means that there are no vertices in either of those j columns outside of the k rows and likewise none in the k rows outside of the j columns.

Claim: The probability that the exposing tree process ends with $2 \leq j+k \leq t+1$ vertices from some starting point tends to zero.

Note that if there are two components, their rows and columns must be disjoint, and hence one of the non-trivial components must have $j+k \leq t$. Thus the claim will complete the proof of the theorem.

Fix $\ell = j+k$. The number of potential exposing trees with $\ell - 1$ vertices in the $t \times t$ grid can be estimated (roughly) as follows. The degrees in the tree can be represented by a sequence of non-negative integers $(a_1, a_2, \ldots, a_\ell)$ with $\sum a_i = \ell - 2$ where a_1, a_2 are the row and column degrees of the first vertex, and a_i is the number of vertices added when the $i-1$st vertex from the queue is explored. The number of such solutions is bounded by $\binom{2\ell-2}{\ell} \leq 4^\ell$. There are fewer than $t^2 \cdot t^{(\ell-2)} = t^\ell$ ways of choosing the vertices that are exposed. Note that this is a rather large over-count: it assumes there are t choices each time, when in reality there is a falling factorial type term and also introduces an ordering when exposing the children of a given vertex. None the less, this upper bounds the number of potential processes for a given ℓ is at most $4^\ell t^\ell$.

Now: for a given one of these potential processes, the $j+k-1 = \ell - 1$ vertices explored must all survive, and the other vertices of their j columns outside of the k rows, and k rows outside of the j columns, must all fail. This has probability $p^{j+k-1}(1-p)^{(t-j)k+(t-k)j} = p^{j+k-1}(1-p)^{t(j+k)-2jk}$. Finally note that $jk \leq \frac{(j+k)^2}{4}$ so that regardless of the individual j, k – for *any* potential process with $\ell = j+k$ fixed the probability of ending is at most

$$p^{j+k-1}(1-p)^{(t-(j+k)/2)\cdot(j+k)} = p^{\ell-1}(1-p)^{(t-\ell/2)\cdot\ell}$$

A union bound over potential exposing trees, shows that the probability that the process ends with a given value of ℓ is at most

$$4^{\ell}t^{\ell}p^{\ell-1}(1-p)^{(t-\ell/2)\cdot\ell} = 4^{\ell}\cdot t((1/2+\varepsilon)\ln(t))^{\ell-1}(1-p)^{t-\ell/2\cdot\ell}$$
$$\leq \exp\left(\ln(t)+\ell\left(\ln(4(1/2+\varepsilon))+\ln\ln(t)-(1/2+\varepsilon)\frac{\ln t}{t}(t-\ell/2)\right)\right).$$

In the last inequlality here, we used the inequality $1-x\leq e^{-x}$ along withe definition of p. Hence, per a union bound over potential ℓ it suffices to show that

$$\sum_{\ell=2}^{t+1}\exp\left(\ln(t)+\ell\left(\ln(4(1/2+\varepsilon))+\ln\ln(t)-(1/2+\varepsilon)\frac{\ln t}{t}(t-\ell/2)\right)\right)\to 0$$

as $t\to\infty$. To do this, we note that for $2\leq\ell\leq 10$, these terms are $o(1)$ individually as for $\ell\leq 10$ $\ell\cdot(1/2+\varepsilon)\cdot\frac{\ln(t)}{t}(t-\ell/2)>(1+\varepsilon/2)\ln(t)$ assuming t is large enough. For $t+1\geq\ell\geq 8$, the dominant part of the terms comes from

$$\ell\cdot(1/2+\varepsilon)\frac{\ln t}{t}\left(\frac{t-1}{2}\right)>(2+\varepsilon/2)\ln(t).$$

Thus these terms are actually $o(t^{-1})$ and as there are fewer than t such terms in total the sum is $o(1)$ as desired. □

4.1 General grids

Theorems 5 and 6 above consider the specialized case where the initial setting is a full $t\times t$ grid. The case of general systems, even the case of general $s\times t$ grids is significantly more complicated. Indeed, in $s\times t$ grids, the asymptotic behavior of how s and t are taken to go to infinity in comparison with one another can give rise to a number of different behaviors, depending on the values of s and t.

For instance, when $s>1$ is fixed, while t goes to infinity the isolation threshold and connectivity threshold differ from each other, and both differ greatly from the above. In this case, we have the following:

Theorem 7. *Consider a full drone system $s\times t$, where drones survive with probability p, where $s>1$ is fixed as $t\to\infty$.*

1. *If $p=\omega(1/t)$, then $\mathbb{P}(\mathcal{I})\to 0$.*
2. *If $p=o(1/\sqrt{t})$, then $\mathbb{P}(\mathcal{C})\to 0$ while if $p=\omega(1/\sqrt{t})$, then $\mathbb{P}(\mathcal{C})\to 1$*

We only sketch the simple proof.

Proof. For (a), the probability a row contains at most one drone is $tp(1-p)^{t-1}+(1-p)^t$ and if $p=\omega(1/t)$ this tends to zero and the result follows from a union bound. For (b), if $p=o(1/\sqrt{t})$ the expected number of columns containing two drones is $\binom{s}{2}\cdot t\cdot p^2\to 0$, which implies the resulting communication graph is disconnected as there will be no communication between rows. Once $p=\omega(1/\sqrt{t})$, each of the $\binom{s}{2}$ pairs of rows will have some column where there is a drone in that column in both rows with high probability, and this forces connectivity. □

When both s and t both tend to infinity, the situation becomes more complicated, and we do not pursue a full investigation here. We do note, however, that the following holds:

Theorem 8. *Consider a full drone system in the $s \times t$ grid, where drones survive with probability p. If $s \leq t$ and $s \to \infty$ then if $p = (1+\epsilon)\frac{\ln(s)}{s}$, then $\mathbb{P}(\mathcal{C}) \to 1$.*

This follows as here an $s \times s$ system contained in the grid is both connected and contains a drone in each row and column.

Acknowledgment. This work was initiated at the IX Spanish Workshop on Geometric Optimization, El Rocío, Huelva, Spain, June 18-22, 2018. We thank the other participants of that workshop - L.E. Caraballo, M.A. Heredia and I. Ventura - for helpful discussions and contribution to a creative atmosphere.

References

1. Bereg, S., Brunner, A., Caraballo, L.E., Díaz-Báñez, J.M., Lopez, M.A.: On the robustness of a synchronized multi-robot system. Journal of Combinatorial Optimization **39**, 988–1016 (2020)
2. Bereg, S., Caraballo, L.E., Díaz-Báñez, J.M., Lopez, M.A.: Computing the k-resilience of a synchronized multi-robot system. Journal of Combinatorial Optimization **36**(2), 365–391 (2018)
3. Camp, T., Boleng, J., Davies, V.: A survey of mobility models for ad hoc network research. Wireless communications and mobile computing **2**(5), 483–502 (2002)
4. Caraballo, L.E., Díaz-Báñez, J.M., Fabila-Monroy, R., Hidalgo-Toscano, C.: Patrolling a terrain with cooperrative uavs using random walks. In: 2019 Int'l Conference on Unmanned Aircraft Systems (ICUAS). pp. 828–837. IEEE (2019)
5. Chung, F., Horn, P.: The spectral gap of a random subgraph of a graph. Internet Math. **4**(2-3), 225–244 (2007)
6. Chung, F., Horn, P., Lu, L.: Percolation in general graphs. Internet Math. **6**(3), 331–347 (2010) (2009)
7. Díaz-Báñez, J.M., Caraballo, L.E., López, M.A., Bereg, S., Maza, J.I., Ollero, A.: A general framework for synchronizing a team of robots under communication constraints. IEEE Trans. Robotics **33**(3), 748–755 (2017)
8. Erdős, P., Rényi, A.: On random graphs. I. Publ. Math. Debrecen **6**, 290–297 (1959)
9. Farinelli, A., Iocchi, L., Nardi, D.: Distributed on-line dynamic task assignment for multi-robot patrolling. Autonomous Robots **41**(6), 1321–1345 (2017)
10. Hayat, S., Yanmaz, E., Muzaffar, R.: Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. IEEE Communications Surveys & Tutorials **18**(4), 2624–2661 (2016)
11. Otto, A., Agatz, N., Campbell, J., Golden, B., Pesch, E.: Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. Networks **72**(4), 411–458 (2018)
12. Yanmaz, E., Yahyanejad, S., Rinner, B., Hellwagner, H., Bettstetter, C.: Drone networks: Communications, coordination, and sensing. Ad Hoc Networks **68**, 1–15 (2018)
13. Yanmaz, E., Costanzo, C., Bettstetter, C., Elmenreich, W.: A discrete stochastic process for coverage analysis of autonomous uav networks. In: 2010 IEEE Globecom Workshops. pp. 1777–1782. IEEE (2010)